3D-based Deep Convolutional Neural Network for Action Recognition with Depth Sequences

Zhi Liu[a], Chenyang Zhang[b], Yingli Tian[b]

[a] *College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400050, China*
[b] *Department of Electrical Engineering, The City College of New York, New York, NY 10031*

Abstract

Traditional algorithms to design hand-crafted features for action recognition have been a hot research area in last decade. Compared to RGB video, depth sequence is more insensitive to lighting changes and more discriminative due to its capability to catch geometric information of object. Unlike many existing methods for action recognition which depend on well-designed features, this paper studies deep learning-based action recognition using depth sequences and the corresponding skeleton joint information. Firstly, we construct a 3D-based Deep Convolutional Neural Network ($3D^2$CNN) to directly learn spatio-temporal features from raw depth sequences, then compute a joint based feature vector named JointVector for each sequence by taking into account the simple position and angle information between skeleton joints. Finally, support vector machine (SVM) classification results from $3D^2$CNN learned features and JointVector are fused to take action recognition. Experimental results demonstrate that our method can learn feature representation which is time-invariant and viewpoint-invariant from depth sequences. The proposed method achieves comparable results to the state-of-the-art methods on the UTKinect-Action3D dataset and achieves superior performance in comparison to baseline methods on the MSR-Action3D dataset. We further investigate the generalization of the trained model by transferring the learned features from one dataset (MSR-

Email addresses: liuzhi@cqut.edu.cn (Zhi Liu), czhang10@ccny.cuny.edu (Chenyang Zhang), ytian@ccny.cuny.edu (Yingli Tian)

Action3D) to another dataset (UTKinect-Action3D) without retraining and obtain very promising classification accuracy.

Keywords: Action Recognition, Deep Learning, Convolutional Neural Network, Depth Sequences, 3D Convolution

1. Introduction

With the ever-increasing growth in the popularity of digital videos, there are many research topics on automatic video analysis. Among these topics, human action recognition (HAR) has been widely applied to a number of real-world applications, *e.g.*, surveillance event detection, human-computer interaction, video retrieval, *etc.* However, it is of great challenge to recognize human actions in unconstrained videos due to some real conditions such as occlusions, different viewpoints, different action speeds, light variances, *etc.* Fortunately, the emergence of depth cameras with acceptable price provides a prospect future for action recognition. Compared with traditional RGB cameras, depth cameras can obtain the conventional two-dimensional (2D) color video sequences as well as the depth sequences which are more insensitive to lighting changes [1] and more discriminative than color and texture features in many computer vision problems such as segmentation, object detection, and activity recognition [2].

According to the difference of extracting features from video sequence, the action recognition methods can be grouped into two categories: hand-crafted feature-based methods and automatic learning feature-based methods. Hand-crafted feature-based methods usually employ a three-stage procedure consisting of feature extracting, feature representation, and classification. Firstly, hand-crafted features such as space-time interest points (STIP) [3], bag-of-visual-words [4, 5], histograms of oriented gradient/histograms of optical flow (HOG/HOF) [6, 7, 8], and motion history image (MHI) [9] are extracted from video sequences. Then more discriminative descriptors are constructed from the extracted features using transformations or clustering techniques, such as Fourier temporal transformation [10, 11] and K-means clustering [12]. Finally, a

classifier is employed on the constructed descriptors to accomplish action recognition. However, it is difficult to generalize the hand-crafted features from one dataset to a different dataset. In 2006, Hinton *et al.* proposed the concept of deep learning to solve the training problem by layer-wise training method [13]. Since then, deep learning has been widely employed in many research areas such as image classification, speech recognition, object recognition, *etc.* [14]. There are also many studies on action recognition using deep learning based on either 2D color image sequences [14, 15, 16, 17] or 3D depth video sequences [18, 19]. However, the inputs of networks in most of these deep learning methods are pre-extracted hand-crafted features instead of raw depth sequences. Unlike the existing methods, in this paper, we apply deep learning to automatically learn discriminative features from raw depth sequences for human action recognition.

We propose a $3D^2$CNN-based framework to automatically learn spatio-temporal feature which we call it high-level feature (Fig.1) from raw depth video sequence. Here, $3D^2$CNN means we take convolution both from spatial and time dimension over the input video by using deep convolutional neural network. To our knowledge, there are little research which use raw depth video sequences as an input in deep learning-based action recognition. Our proposed framework is evaluated on two well-known datasets: UTKinect-Action3D [20] and MSR-Action3D [12]. Our method obtains comparable performance to state-of-the-art methods on the UTKinect-Action3D dataset (Table 4) and achieves superior performance in comparison to baseline methods on the MSR-Action3D dataset (Table 5).

The key contributions of this work are summarized as follows:

(1) We propose a novel $3D^2$CNN-based framework for action recognition from depth video sequences. This framework can automatically extract spatio-temporal features from raw depth video sequences.

(2) We evaluate the influence of different sizes of input sequences to performance of feature learning of $3D^2$CNN, it shows that $3D^2$CNN can learn general structure information of the video for action recognition. The classification

accuracies are slightly improved and tend to be stable with larger spatial sizes.

(3) We investigate the generalization of the trained model by transferring the learned hierarchical feature representations from one dataset (MSR-Action3D) to a different dataset (UTKinect-Action3D) without retraining. Experimental performance demonstrates the high generalization performance of transferring the trained model to different datasets.

(4) Without designing complex hand-crafted features, our framework achieves comparable results as the state-of-the-art methods.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 presents the overall structure of the proposed framework and the detail of $3D^2$CNN model. The experimental results and discussions are described in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Hand-Crafted Feature-based Action Recognition

Generally, feature extraction from video sequences is one of the most important steps in HAR. Over the past years, low-level features, such as scale-invariant feature transform (SIFT) [21], STIP [22, 23], HOG/HOF [8] and speeded up robust features (SURF) [24], have been successfully employed in traditional RGB video-based activity recognition. Wang and Schmid [25] proposed an action recognition method with improved trajectories. They obtain motion Vectors independent of camera motion by matching feature points between frames using SURF descriptors and dense optical flow, and improve motion-based descriptors significantly. Shao *et al.* [26] fused different feature representations for action recognition using a novel spectral coding algorithm called kernelized multiview projection (KMP). However, RGB video-based action recognition tasks invariably show low performance owing to its incapability to catch 3D information for each frame. Fortunately, the recent emergence of cost-effective depth sensors

such as Kinect [1], attracted more attentions from researchers to reconsider visual tasks such as human attribute recognition, video segmentation, and activity recognition by using depth information as input instead of RGB images. Ye *et al.* [18] and Han *et al.* [27] summarized a detailed survey on HAR from depth camera. Among depth-based HAR methods, HOG [6, 7], STIP [22, 3, 23],bag-of-3D points [12, 4], skeleton joints [28, 11, 20] are mostly used features. Ni *et al.* [29] presented a novel idea for activity recognition by applying depth based filters to remove false detection and then combining data from conventional camera and a depth sensor. Ye *et al.* [30] proposed an algorithm for creating free-viewpoint video of interacting humans using three hand-held Kinect cameras by estimation of human poses and camera poses. Yang *et al.* [6] proposed an efficient feature representation by projecting depth maps onto three orthogonal planes to generate the depth motion maps (DMM) which accumulate global activities through an entire video sequence. HOG is then computed from DMM as the representation of each action video. Oreifej and Liu [7] used a histogram to represent the distribution of the surface normal orientation in the 4D space of time, depth, and spatial coordinates in order to capture the complex joint shape motion cues at pixel-level. Li *et al.* [12] utilized a bag of 3D points to characterize a set of salient postures which is used as the nodes in the action graph. Roshtkhari and Levine [4] constructed a codebook of spatio-temporal video volumes which then is assembled to a large contextual volume in order to train a probabilistic model of video volumes and the spatio-temporal compositions. In [28], Zanfir *et al.* proposed a moving pose descriptor by using the configuration, speed, and acceleration of joints. Yu *et al.* [31] proposed a hand-crafted feature called Local Flux Feature (LFF). Then the LFFs from RGB and depth channels are fused into a Hamming space via the Structure Preserving Projection (SPP). Their experiments show the effectiveness of combining LFFs from RGB-D channels via SPP. Vemulapalli *et al.* [11] used skeleton joint information to model the 3D geometric relationships between various body parts. In this way, human actions can be modeled as curves in a Lie group [32]. Different from [11], Xia *et al.* [20] constructed histograms of 3D joint locations (HOJ3D) as a compact

representation of postures. In [22], the authors formed various STIP features by combining different interest point detectors and conducted comparison with some classical STIP methods. Their experiments revealed that skeleton joint information can greatly improve performance by fusing spatio-temporal features and skeleton joints for depth-based action recognition [22]. In addition, other proposed representations of depth sequences include depth motion map [33], super normal vector (SNV) [2], actionlet [10], spatio-temporal depth cuboid [34], cloud points [35], *etc.*.

2.2. Machine Learning-based Action Recognition

All methods mentioned above utilize modern feature extraction or hand-crafted features which are achieving remarkable performance. However, these features, such as MHI [9], HOG/HOF [8] and HOG3D [36], are hand-crafted with respect to specific video sequences and specific applications. Therefore, they are difficult to be generalized to other real-world scenarios because it is difficult to know which features are important to the recognition tasks without retraining. Furthermore, with respect to hand-crafted methods, feature extraction and classifier training are conducted sequentially and individually. Thus, the training error of each sample is hard to be back propagated to the entire pipeline to improve feature extraction. Instead, machine learning methods, and in particular convolutional neural networks (CNN) [37] can provide some solutions. The machine learning methods have achieved very good performance on many visual tasks, from image classification [17], pedestrian detection [38] to pose recognition [1] and image de-noising [39]. Moreover, Donahue *et al.* [40] demonstrated that features extracted from the deep convolutional network trained on large datasets are generic and benefit to other visual recognition problems. In summary, machine learning methods show more and more superiority in computer vision and image processing tasks, which also promote more and more studies on machine learning-based action recognition. Jin *et al.* [41] did very interesting work to combine hand-crafted features and machine-learned features for RGB-D object recognition. While Liu *et al.* [42, 43] presented us

a good idea that Genetic Programming is used to learn discriminate spatio-temporal features from RGB-D video. They filtered a set of 3D operators which were then automatically assembled into a dynamic-depth tree-structured chain to extract the features from the input RGB-D data. Ji *et al.* [14] proposed a 3D-CNN model applied into RGB-based action recognition. They first applied a set of hardwired kernels to generate multiple channels of information from the input frames, and then captured the motion information encoded in multiple adjacent frames by performing 3D convolutions, and the final feature representation combines information from all channels. Wu and Shao [44] extracted features from skeletal joint information and applied deep neural network to predict probability distribution over the state of HMMs for action recognition. Le *et al.* [15] proposed a method by combining advantages of independent subspace analysis (ISA) and CNN. They utilized ISA to learn invariant spatio-temporal features which are then used to learn higher or more abstract level representation by stacked CNN. Similar to [15], Lin *et al.*[16] adopted the same framework as [15]. The only difference is that [15] used ISA, while Lin *et al.*[16] employed slow feature analysis (SFA). ISA aims at learning feature representations tend to be invariant to spatial translation of stimuli, while SFA is intended to find a more stable and slowly varying feature representation of video sequences. The methods mentioned above extract features first and then use stacked CNN to learn higher level representations. Molchanov *et al.* [45] collected a gesture data set and obtained 77.5% recognition accuracy using deep 3D Convolutional neural network. Different from these methods, paper [46] employed CNN to automatically learn features by taking raw RGB video sequences as input without any hand-crafted feature extracting process. Their CNN inputs are two separate streams of processing, i.e. a context stream which captures the whole structure information and a high-resolution stream which captures fine-grained information of an action. In [19], Valle and Starostenko described a simple survey on human action recognition and employeed a 2D CNN-based deep learning method to discriminate walking from running for RGB sequences. Tran *et al.* [47] proposed 3D ConvNets to learn spatio-temporal feature using deep 3-dimensional

convolutional networks trained on a RGB video dataset.

Machine learning-based methods have achieved great performance in action recognition tasks. However, most of them are based on RGB-based video sequences which is inferior to catch more discriminative information relative to depth sequences. What's more, due to the difficulty of designing deep neural network of 3D input in earlier years, most of machine learning methods just take deep learning as a way of dimension reduction. There is little research that takes raw depth video sequences as the input of deep neural network to study HAR. In this paper, we proceed along this direction. We fuse the learned features from raw depth sequence and the JointVector calculated from skeleton joints to accomplish action recognition. First we develop a $3D^2$CNN to automatically learn spatio-temporal features from raw depth sequences. Then we calculate a JointVector for each sequence by taking into account the position and angle information of skeleton joints. Finally, the SVM classification results from high-level feature and JointVector are fused to recognize actions.

3. Proposed $3D^2$CNN-based Framework

Fig.1 presents the overall structure of $3D^2$CNN for action recognition. Our HAR pipeline mainly consists of three parts, i.e. $3D^2$CNN-based feature learning, JointVector calculating, and fusion of classification results. Feature learning phase learns the spatio-temporal features from raw depth videos with a deep neural network. In our work, we just take the output of the first full connected layer as the extracted feature from network (Fig.2). JointVector calculating phase obtains a JointVector according to skeleton joints information for each depth video sequence (Fig.3). Finally, the SVM classification results from high-level feature and JointVector are fused to execute action recognition. For fusion process, we just simply sum the probability of corresponding action from two input classifiers with different fusing weights, and the maximum one will be the recognized action. In this paper, the weights of high-level feature and JointVector classifiers are set to 5 and 3 respectively according to our experiment.

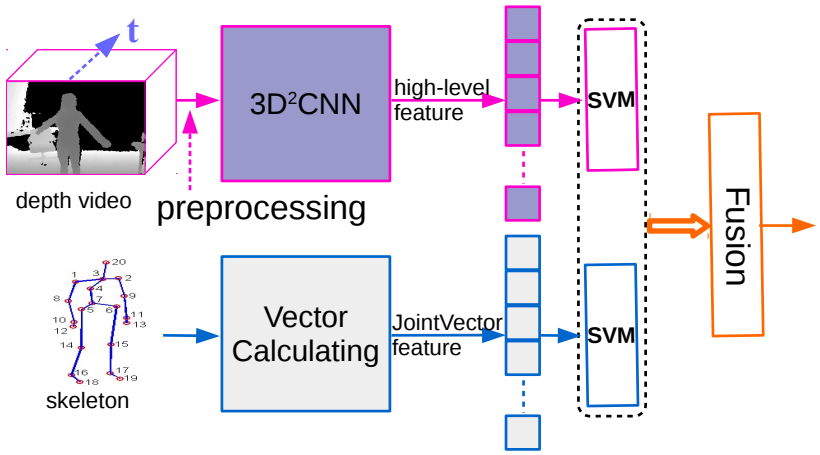

Figure 1: The overall structure of the $3D^2$CNN-based HAR Framework. $3D^2$CNN learns high-level features from depth video and Vector Calculating component calculates joint feature (JointVector) according to joint skeleton. The SVM classification results from high-level feature and JointVector are fused to recognize action.

3.1. Learning High-level Feature from Raw Depth Videos

The 3D deep network for learning features in Fig.2 is responsible for high-level feature learning tasks. We construct the 3D deep network according to previous studies [48] and the size of our training data set. It consists of two 3D convolution layers, each of which followed by a 3D max pooling layer. The input to the network would be cuboids which originate from raw depth sequences after being simply preprocessed and normalized to the same size for a specific dataset. Because the lengths of action videos may vary in different datasets, it is difficult to normalize all the videos to one size while preserve enough information of the videos. Therefore different datasets may have different cuboid sizes as input of the network. Here, the framework in Fig.2 takes the MSR-Action3D dataset as an example. First, the normalized depth cuboid (action video) of size $height \times width \times time$ ($32 \times 32 \times 38$) is input to the deep neural network after raw depth sequence being simply preprocessed. The input cuboid is processed by the first 3D convolution layer (CL1) and 3D max pooling layer (MP1). And then the second 3D convolution layer (CL2) and 3D max pooling layer (MP2)

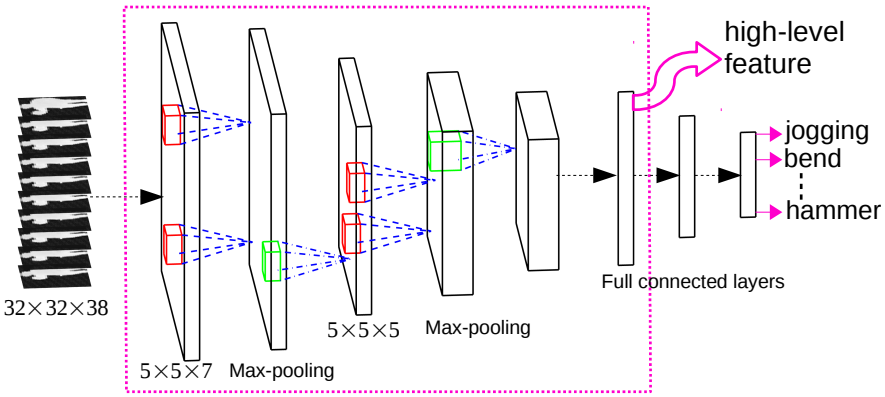

Figure 2: 3D Deep Convolutional Neural Network. Depth sequences are input to the network after being normalized to fixed size cuboids. Then execute two 3D convolutional layers which are followed by three full connected layers and a SVM classifier.

are following layers. Finally, three fully connected linear transformation layers (FCL) are applied and soft-max layer is used as the classifier. The kernel size of CL1 and CL2 are $5 \times 5 \times 7$ (5×5 in the spatial dimension and 7 in the temporal dimension) and $5 \times 5 \times 5$ respectively. In the subsequent 3D max pooling layer after convolution layer, we apply $2 \times 2 \times 2$ sub sampling on both MP1 and MP2 layers, which leads to a reduced spatial and temporal resolution. The numbers of feature maps (i.e. kernels) of CL1 and CL2 are 32 and 128 respectively. The vector dimensions of three full connected linear transformation are 2056, 512, and 128 respectively. While for the UTKinect-Action3D dataset, the input size is $32 \times 32 \times 28$. The kernel sizes of both CL1 and CL2 are $5 \times 5 \times 5$. Table 1 list the size of convolutional filter and convolution stride corresponding to cuboids of different input size. Unless stated specifically, all other parameters in following experiments are same as that used on the MSR-Action3D dataset.

3.2. JointVector Calculation

Here we propose a straightforward method to calculate JointVector for each sequence. The datasets provide the skeleton information of 20 joints which are *Hip center, Spine, Center between shoulders, Head, Left shoulder, Left elbow,*

Table 1: Parameters setting of $3D^2$CNN corresponding to cuboids of different input size. Here, Size32 means the cuboid size is $32 \times 32 \times 28$ for UTKinect-Action3D and is $32 \times 32 \times 38$ for MSR-Action3D, and so on. Stride setting is for both CL1 and CL2.

	UTKinect-Action3D			MSR-Action3D		
	Size32	Size64	Size128	Size32	Size64	Size128
CL1	$5 \times 5 \times 5$	$6 \times 6 \times 5$	$6 \times 6 \times 5$	$5 \times 5 \times 7$	$6 \times 6 \times 7$	$6 \times 6 \times 7$
CL2	$5 \times 5 \times 5$	$5 \times 5 \times 5$	$5 \times 5 \times 5$	$5 \times 5 \times 5$	$5 \times 5 \times 5$	$5 \times 5 \times 5$
Stride	$1 \times 1 \times 1$	$2 \times 2 \times 1$	$2 \times 2 \times 1$	$1 \times 1 \times 1$	$2 \times 2 \times 1$	$2 \times 2 \times 1$

Left wrist, Left hand, Right shoulder, Right elbow, Right wrist, Right hand, Left hip, Left knee, Left ankle, Left foot, Right hip, Right knee, Right ankle and *Right foot*. JointVector is formulated by concatenating two independent feature vectors. One is the pairwise relative position vector presented in paper [10]. Another is *Hip center* based vector (HCBV) proposed in this paper. Since each skeleton information contains x, y and z coordinate information, we can calculate the distance and angle information of every joint relative to a designated joint. Fig.3 presents the calculation procedure of HCBV. Our HCBV calculation method takes the *Hip center* joint as the original point of the 3D coordinate because it is the steadiest joint compared to other joints. Thus for each joint in addition to the *Hip center* joint, we calculate the following three parameters: distance to origin (d), angle of elevation (ϕ) and Azimuthal angle (θ). As we know, each frame has 19 joints except the *Hip center* joint. Therefore for a depth sequence with tNum frames, we can obtain a $3 \times 19 \times tNum$ HCBV by concatenating the three parameters of all joints of all frames in the depth sequence.

3.3. Model Implementation

At the feature learning step, the $3D^2$CNN model is implemented in Torch7 [49] which is a scientific computing framework based on LuaJIT with wide support for machine learning algorithms. The negative log likelihood criterion is used in the optimization phase, which requires the outputs of the trainable

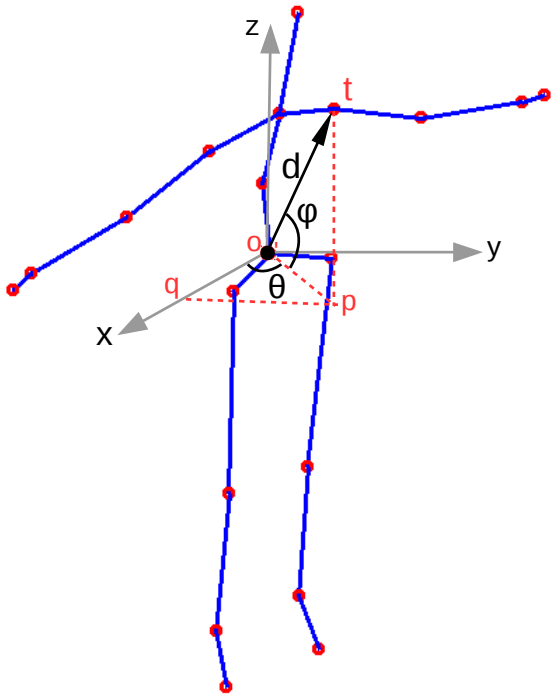

Figure 3: Calculating joint-center based vector (HCBV) for a frame. *Hip center* of each frame is set as the origin of 3D coordinates. Then distance to origin (d), angle of elevation (ϕ) and Azimuthal angle (θ) are calculated for each joint in every frame.

model to be properly normalized log-probabilities and can be achieved using a soft max function. We employ a stochastic gradient algorithm to train the neural networks. The active function is the hyperbolic tangent function (tanh) since our datasets are relatively small and the learning rate is set to 5e-4 empirically. On classification phase, the linear classifier (LIBLINEAR) [50] with default parameters is used for action recognition.

4. Experimental Results

4.1. Datasets

In this section, we present a comparative performance evaluation of our proposed method on two datasets: the UTKinect-Action3D dataset [20] and the MSR-Action3D dataset [12]. Both two datasets were captured using a stationary Kinect sensor and provide the 3D locations of 20 joints. The UTKinect-Action3D dataset consists of 10 actions performed by 10 different subjects. The 10 actions are *Walk*, *Sit down*, *Stand up*, *Pick up*, *Carry*, *Throw*, *Push*, *Pull*, *Wave hands* and *Clap hands*. Each subject performed every action twice. There are total of 199 effective action sequences. For convenience, we use 200 action sequences in our experiments by filling the missing action *Carry* of the second performance of the 10th subject using frames from No.1242 to No.1300. The challenges in the UTKinect-Action3D are viewpoint variations and high intra-class variations. The MSR-Action3D dataset consists of 20 actions performed by 10 different subjects. Each subject performed every action two or three times. In total, there are 567 action sequences. According to the test setting of the baseline method [12], the 20 actions was divided into action subsets AS1, AS2 and AS3 (Table 2), each consisting of 8 actions. The AS1 and AS2 were intended to group actions with similar movement, while AS3 was intended to group complex actions together. The main challenge in the MSR-Action3D dataset is that some of the actions are very similar to each other, especially in AS1 and AS2. For instance, action *Hammer* is likely to be confused with *Forward punch* in AS1 and action *Draw x* is a little different from action *Draw circle* only in the part

Table 2: The three subsets of actions used in the experiments.

AS1	AS2	AS3
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

of right hand in AS2. In order to make our proposed method less insensitive to different subjects, we conduct a preprocessing for each video sequence including foreground extraction, person-centered bounding box detection, spatial and temporal dimension normalization, and depth value normalization. After foreground extraction, only the depth values of foreground pixels are kept and values of other pixels are set to zero. In bounding box cutting, the bounding box of the person is cropped for every frame in an action video and the largest bounding box is used as the video's bounding box. The spatial and temporal dimension normalization is applied to resize all action videos in a dataset to cuboids of designated size. While the depth value normalization takes min-max normalization of depth values of all pixels to 0-1 range from the video level. Fig.4 presents data preprocessing pipeline of our methods.

4.2. Performance Evaluation on the UTKinect-Action3D Dataset

We first evaluate the efficiency of the proposed $3D^2$CNN method and compare the recognition result with the state-of-the-art results on the UTKinect-Action3D and the MSR-Action3D datasets. For the UTKinect-Action3D dataset, we train a network for each subject. For each network, sequences of one subject are used for testing and sequences of other subjects are used for training. Here, we call it *leave one subject out cross validation* which is hasher than the

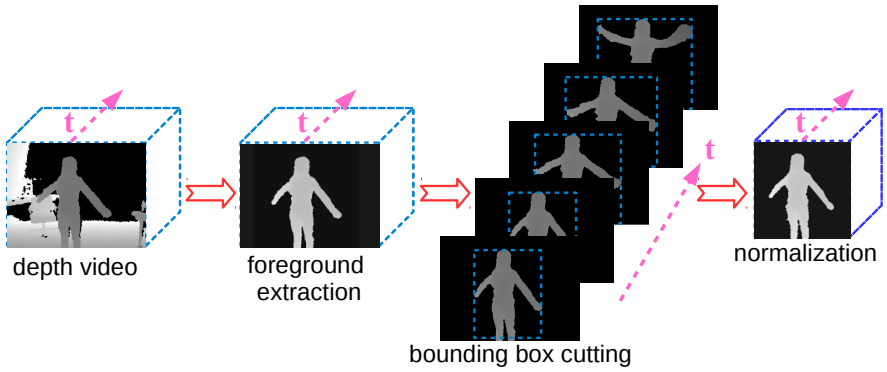

Figure 4: Preprocess pipeline of depth sequences. For each dataset, four preprocessing steps are conducted for all depth videos including foreground extraction, bounding box cutting, normalizing of height, width and frame numbers, and depth value normalization.

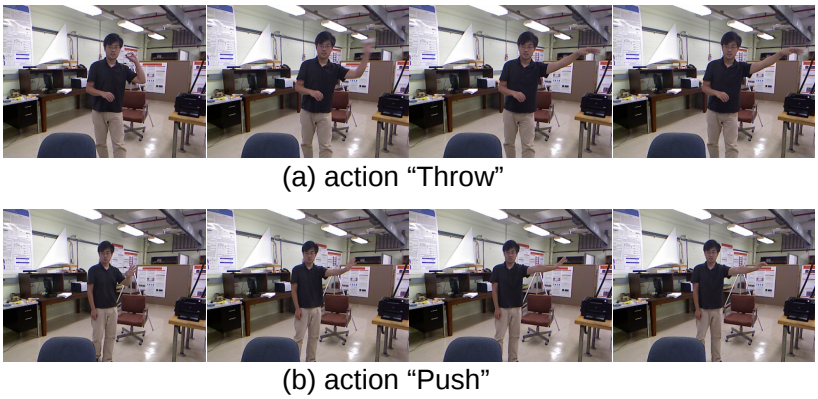

Figure 5: Two similar action sequences in UTKinect-Action3D dataset.

experiment setup in paper [20] which used *leave one sequence out cross validation* (LOOCV). The recognition accuracies for each action are listed in Table 3. From Table 3, we observe that most of the actions are correctly classified and the average accuracy is 95.5%. The action *Throw* obtains the lowest accuracy while most of the wrong classified samples are confused with action *Push* due to the similarity of the two actions (Fig.5). Table 4 shows the performance of our method compared to the state-of-the-art approaches. Here, we adopt cross-subject experimental setting in which subjects 1,3,5,7,9 are used for training and subjects 2,4,6,8,10 are used for testing. Our results are obtained

Table 3: The performance of each action on the UTKinect-Action3D dataset(Average:95.5%).

Action	walk	sit down	stand up	pick up	carry
Accuracy	100	100	100	95	90
Action	throw	push	pull	wave hand	clap hand
Accuracy	70	95	100	100	100

when the input to $3D^2$CNN are cuboids of size $128 \times 128 \times 28$. The action recognition accuracies are 82% by only using $3D^2$CNN and 93% by only using JointVector. By fusion of $3D^2$CNN and JointVector, our proposed method obtains accuracy of 96% which is comparable to the state-of-the-art result of [11]. It is worthwhile to note that [11] prepared 10 training sets and testing sets which are used in their cross-subject experiments. By running their code in our cross-subject setting, their method achieves an accuracy of 95.96% which is a little worse than our results. Furthermore, our proposed method is much faster than that of [11]. The average time consumption of extracting $3D^2$CNN learned features and calculating JointVector for a sequence is 1.18 seconds while [11] needs about 6.53 seconds on the same computer who has 4 processor of Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz and total memory of 4G. Thus, we conclude that the features learned from $3D^2$CNN are complementary with the JointVector features in action recognition. Since the main challenge in the UTKinect-Action3D is the viewpoint variation, the high recognition rate demonstrates that our proposed method can learn view invariant features. Compared to [11], our method is straightforward by combining the $3D^2$CNN learned features and simple JointVector.

4.3. Performance Evaluation on the MSR-Action3D Dataset

For the MSR-Action3D dataset, we followed the same test setting of the baseline method [12] in which the dataset was divided into action subsets AS1, AS2 and AS3 (Table 2), each consisting of 8 actions. For each action subset, three different division methods of training and testing sets are performed. They are *Test One*, *Test Two* and *Cross Subject*. In *Test One*, random 1/3 of the

Table 4: The performance of our method on the UTKinect-Action3D dataset, compared to the state-of-the-art approaches

Method	Accuracy
Xia *et al.* (2012)[20]	90.92%
Devanne *et al.* (2013)[51]	91.5%
Chrungoo *et al.* (2014)[52]	91.96%
Vemulapalli *et al.* (2014)[11]	**97.08%**
$3D^2$CNN	82%
JointVector	93%
Proposed	96%

samples were used for training and the rest for testing. In *Test Two*, random 2/3 samples were used as training samples and the rest for testing. In *Cross Subject*, half of the subjects were used for training and the rest subjects were used for testing. Our proposed method is compared with two state-of-the-art approaches which also present all experiment results on subsets AS1, AS2 and AS3. As shown in Table 5, we observe that the performance of our proposed algorithm is a little worse than [12] and [20] on AS1 and AS2, and is much better than them on AS3. The reason may be that AS1 and AS2 group actions with similar movement, while AS3 groups complex actions together. So the movements of AS3 contain more global information than that of AS1 and AS2. From the conclusion of Section 4.4, the superiority of $3D^2$CNN is to learn the structural information of the video, not the fine-grained one. While hand-crafted methods extract features from every pixel, which means structural knowledge and fine-grained features are treated in the same way.

4.4. Performance of Different Spatio-temporal Size

In order to evaluate the performance on different spatio-temporal size to performance of $3D^2$CNN and proposed method, we normalize the depth sequences of each action to three different sizes. For the UTKinect-Action3D dataset, the cuboid size of $32 \times 32 \times 28$, $64 \times 64 \times 28$ and $128 \times 128 \times 28$ are evaluated. While

Table 5: Recognition results of our algorithm on the MSR-Action3D dataset. In *Test One*, random 1/3 of the samples were used as training samples and the rest as testing samples. In *Test Two*, random 2/3 samples were used as training samples. In *Cross Subject* test, half of the subjects were used as training and the rest of the subjects were used as testing (%).

	Test One			Test Two			Cross Subject		
	[12]	[20]	Our	[12]	[20]	Our	[12]	[20]	Our
AS1	89.5	**98.47**	92.03	93.4	**98.61**	92.78	72.9	**87.98**	86.79
AS2	89.0	**96.67**	88.59	92.9	**97.92**	97.06	71.9	**85.48**	76.11
AS3	**96.3**	93.47	95.54	96.3	94.93	**98.59**	79.2	63.45	**89.29**
Avg	91.6	**96.20**	92.05	94.2	97.15	**98.14**	74.7	78.97	**84.07**

Table 6: The performance comparison on the UTKinect-Action3D dataset on different spatio-temporal sizes.

	UtkCross32	UtkCross64	UtkCross128
$3D^2$CNN	81%	82%	82%
Proposed	95%	96%	96%

for the MSR-Action3D dataset, we use the input cuboids of size $32 \times 32 \times 38$, $64 \times 64 \times 38$ and $128 \times 128 \times 38$. Then we evaluate their performances using cross-subject method on all actions, in which subjects (1,3,5,7,9) are used for training and subjects (2,4,6,8,10) are used for testing for both datasets. Table 6 and 7 report the performance comparison on the UTKinect-Action3D and the MSR-Action 3D AS3 dataset using different spatial size respectively. From these two tables, we observe that the classification accuracy is slightly improved and tends to be stable with larger spatial sizes since $3D^2$CNN can learn the whole structure information of the video. The video reserves majority of information as long as the cuboids retain basic structure of each frame and keep smooth in time dimension after being resized.

4.5. Transferring the Learned High-level Feature to Different Dataset

Compared to hand-crafted feature-based methods, deep learning-based methods are deemed to be more generic in transferring of learned model among differ-

Table 7: The performance comparison on the MSR-Action3D AS3 dataset on different spatio-temporal sizes

	MsrCross32	MsrCross64	MsrCross128
$3D^2$CNN	88.16%	89.47%	88.16%
Proposed	89.29%	90.18%	90.18%

ent datasets. Oquab *et al.* [53] showed that how image representations learned with CNNs on large-scale ImageNet datasets can be efficiently transferred to other visual recognition tasks with limited amount of training data. In the paper [46], Karpathy *et al.* found that the motion features learned on the Sports-1M dataset can be generalized to other datasets and class categories, such as on the UCF-101 Activity Recognition dataset. Here, we evaluate the transferring generalization of our trained $3D^2$CNN model with different depth action recognition datasets. First, we train the $3D^2$CNN model using the MSR-Action3D dataset with input cuboid size is $32 \times 32 \times 38$, and then test the performance on the UTKinect-Action3D dataset without retraining. Before testing on the UTKinect-Action3D dataset, the sequences of size $32 \times 32 \times 28$ are scaled to $32 \times 32 \times 38$ in order to let the input cuboids be fit to the trained $3D^2$CNN model. There are nearly no overlap actions in these two datasets. In this way, we can extract high-level features (Fig.2) for each video in the UTKinect-Action3D dataset. Then, the extracted high-level features and JointVector are fused to recognize the actions. In this feature transferring experiment, we achieve an excellent performance at an accuracy of 95%. The performance is comparable to the state-of-the-art results of paper [11] whose recognition accuracy is 97.08%. Transferring experiment show that $3D^2$CNN can learn more nature features of the video and is more generic than hand-crafted feature-based methods.

5. Conclusions

In this paper, we have proposed an action recognition framework which includes three components, i.e., $3D^2$CNN, JointVector calculation, and classifiers

fusion. Firstly, we extract high-level features from depth video with the deep neural networks. Then a JointVector is calculated for each depth sequence according to skeleton joints information. Finally, the SVM classification results from high-level features and JointVector are fused for action recognition. Torch7 [49] is employed to implement our $3D^2$CNN model and LIBLINEAR [50] is used for classification. The proposed method is very straightforward by automatically learning high-level features from raw depth sequence with the fusion of JointVector from skeleton information for each sequence. Our proposed framework has been evaluated on the MSR-Action3D dataset and the UTKinect-Action3D dataset and has achieved comparable results as the state-of-the-art methods. Furthermore, we have evaluated the generalization of the trained model using different datasets in training and testing. Experimental results demonstrate that the trained model on one depth video sequence dataset can be transferred to different datasets.

6. Acknowledgement

This work was supported in part by NSF grants EFRI-1137172, IIS-1400802, Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJ1400926) and Chongqing National Science Foundation (cstc2013jcyjA40038).

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 1297–1304.

[2] X. Yang, Y. Tian, Super normal vector for activity recognition using depth sequences, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 804–811.

[3] X. Peng, Y. Qiao, Q. Peng, Motion boundary based sampling and 3d co-occurrence descriptors for action recognition, Image and Vision Computing 32 (9) (2014) 616–628.

[4] M. J. Roshtkhari, M. D. Levine, Human activity recognition in videos using a single example, Image and Vision Computing 31 (11) (2013) 864–876.

[5] S. O'Hara, Y. M. Lui, B. A. Draper, Using a product manifold distance for unsupervised action recognition, Image and vision computing 30 (3) (2012) 206–216.

[6] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: Proceedings of the 20th ACM International Conference on Multimedia, MM '12, ACM, New York, NY, USA, 2012, pp. 1057–1060.

[7] O. Oreifej, Z. Liu, Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 716–723.

[8] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1–8.

[9] J. W. Davis, A. F. Bobick, The representation and recognition of human movement using temporal templates, in: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, IEEE, 1997, pp. 928–934.

[10] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 1290–1297.

[11] Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3D skeletons as points in a lie group, in: Computer Vision and

Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 588–595.

[12] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, 2010, pp. 9–14.

[13] G. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation 18 (7) (2006) 1527–1554.

[14] V. Shuiwang Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 221–231.

[15] Q. V. Le, W. Y. Zou, S. Y. Yeung, A. Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3361–3368.

[16] S. Lin Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, S. Yan, Dl-sfa: Deeply-learned slow feature analysis for action recognition, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 2625–2632.

[17] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[18] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A survey on human motion analysis from depth data, in: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, Springer, 2013, pp. 149–187.

[19] A. Valle, O. Starostenko, Recognition of human walking/running actions based on neural network, in: Electrical Engineering, Computing Science and Automatic Control (CCE), 2013 10th International Conference on, 2013, pp. 239–244.

[20] L. Xia, C.-C. Chen, J. K. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, pp. 20–27.

[21] D. G. Lowe, Object recognition from local scale-invariant features, in: Computer vision, 1999. The proceedings of the seventh IEEE international conference on, Vol. 2, Ieee, 1999, pp. 1150–1157.

[22] Y. Zhu, W. Chen, G. Guo, Evaluating spatiotemporal interest point features for depth-based action recognition, Image and Vision Computing 32 (8) (2014) 453–464.

[23] T. H. Thi, L. Cheng, J. Zhang, L. Wang, S. Satoh, Structured learning of local features for human action classification and localization, Image and Vision Computing 30 (1) (2012) 1–14.

[24] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Computer vision and image understanding 110 (3) (2008) 346–359.

[25] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 3551–3558.

[26] L. Shao, L. Liu, M. Yu, Kernelized multiview projection for robust action recognition, International Journal of Computer Vision (2015) 1–15.

[27] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with microsoft kinect sensor: A review, Cybernetics, IEEE Transactions on 43 (5) (2013) 1318–1334.

[28] M. Zanfir, M. Leordeanu, C. Sminchisescu, The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection, in: Computer Vision (ICCV), 2013 IEEE International Conference on, 2013, pp. 2752–2759.

[29] B. Ni, Y. Pei, P. Moulin, S. Yan, Multilevel depth and image fusion for human activity detection, Cybernetics, IEEE Transactions on 43 (5) (2013) 1383–1394.

[30] G. Ye, Y. Liu, Y. Deng, N. Hasler, X. Ji, Q. Dai, C. Theobalt, Free-viewpoint video of human actors using multiple handheld kinects, Cybernetics, IEEE Transactions on 43 (5) (2013) 1370–1382.

[31] M. Yu, L. Liu, L. Shao, Structure-preserving binary representations for rgb-d action recognition, EEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

[32] R. M. Murray, Z. Li, S. S. Sastry, S. S. Sastry, A mathematical introduction to robotic manipulation, CRC press, 1994.

[33] C. Zhang, Y. Tian, Edge enhanced depth motion map for dynamic hand gesture recognition, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on, 2013, pp. 500–505.

[34] L. Xia, J. K. Aggarwal, Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 2834–2841.

[35] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3d action recognition with random occupancy patterns, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 872–885.

[36] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association, 2008, pp. 275–1.

[37] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural computation 1 (4) (1989) 541–551.

[38] Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 3626–3633.

[39] H. C. Burger, C. J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with bm3d?, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 2392–2399.

[40] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, arXiv preprint arXiv:1310.1531.

[41] L. Jin, S. Gao, Z. Li, J. Tang, Hand-crafted features or machine learnt features? together they improve rgb-d object recognition, in: Multimedia (ISM), 2014 IEEE International Symposium on, IEEE, 2014, pp. 311–319.

[42] L. Liu, L. Shao, Learning discriminative representations from rgb-d video data, in: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, AAAI Press, 2013, pp. 1493–1500.

[43] L. Liu, L. Shao, X. Li, K. Lu, Learning spatio-temporal representations for action recognition: A genetic programming approach, IEEE TRANSACTIONS ON CYBERNETICS.

[44] D. Wu, L. Shao, Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 724–731.

[45] P. Molchanov, S. Gupta, K. Kim, J. Kautz, Hand gesture recognition with 3d convolutional neural networks, 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops.

[46] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in:

Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 1725–1732.

[47] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4489–4497.

[48] D. Hunter, H. Yu, M. S. Pukish III, J. Kolbusz, B. M. Wilamowski, Selection of proper neural network sizes and architectures, a comparative study, Industrial Informatics, IEEE Transactions on 8 (2) (2012) 228–240.

[49] R. Collobert, K. Kavukcuoglu, C. Farabet, Torch7: A matlab-like environment for machine learning, in: BigLearn, NIPS Workshop, 2011.

[50] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.

[51] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, Space-time pose representation for 3d human action recognition, in: New Trends in Image Analysis and Processing–ICIAP 2013, Springer, 2013, pp. 456–464.

[52] A. Chrungoo, S. Manimaran, B. Ravindran, Activity recognition for natural human robot interaction, in: Social Robotics, Springer, 2014, pp. 84–94.

[53] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 1717–1724.